Open Solutions to Distributed Control in Ground Tracking Stations

by
Wm. Randy Heuser
Member of the Technical Staff
Jet Propulsion Laboratory
California Institute of Technology
Mail Stop 301-235
4800 Oak Grove Drive
Pasadena, California 91109
Office phone (818) 354-0956, Fax (818) 354-9068
email: rheuser@binky.jpl.nasa.gov

Abstract

The advent of high speed local area networks has made it possible to interconnect small, powerful computers to function together as a single large computer. Today, distributed computer systems are the new paradigm for large scale computing systems. However, the communications provided by the local area network is only one part of the solution. The services and protocols used by the application programs to communicate across the network are as indispensable as the local area network. And the selection of services and protocols that do not match the system requirements will limit the capabilities, performance and expansion of the system. Proprietary solutions are available but are usually limited to a select set of equipment. However, there are two solutions based on "open" standards. The question that must be answered is "which one is the best one for my job?"

This paper examines a model for tracking stations and their requirements for inter-processor communications in the next century. The model and requirements are matched with the model and services provided by the five different software architectures and supporting protocol solutions. Several key services are examined in detail to determine which services and protocols most closely match the requirements for the tracking station environment. The study reveals that the protocols are tailored to the problem domains for which they were originally designed. Further, the study reveals that the process control model is the closest match to the tracking station model.

Introduction

Tracking stations are a collection of different pieces of equipment, integrated into a single system to support communications between the ground and a spacecraft. The antenna equipment, the receiver equipment, the transmitting equipment and associated signal processing equipment are built by experts in their field. Over the past decade, computers have been incorporated into this equipment to operate and automate their increasingly complex functions. Today, this computerized equipment (called subsystems) can be linked together with communication protocols into an operating tracking station. However, the degree of difficulty to integrate these subsystems into a single tracking station, and the level of automation that can be achieved, will be a direct function of the protocol selected. This paper examines a number of non-proprietary protocols that have been used or suggested as possible candidates for the tracking station application.

Today, commercial vendors market computer controlled components for tracking stations. As government budgets shrink and commercial exploitation of space grows, these products offer cost effective solutions to one-of-a-kind development efforts. However, vendors are looking to protect their share of the market and their proprietary products. To this end, some vendors offer complete, fully automated tracking stations. However, these turn-key solutions usually provide limited services. And in general, single vendor solutions are not attractive to government or industry. An "open solution" provides a multi-vendor environment where the best products for the job can be integrated into a single system. Commercial inter-processor communications protocols that provide an "open solution" while affording protection to proprietary products are needed to support the integration of different vendor components into a single automated tracking station.

Operational Scenario

An examination of the various candidate protocols is facilitated with a simple model of a tracking station. Consider the construction of a new tracking station to be built using commercial-off-the-shelf components. Four different companies will provide computer controlled equipment that will be integrated into a fully automated tracking station. The elements include: the antenna subsystem, the receiver subsystem, the telemetry subsystem and the command subsystem (see Figure 1). Each subsystem is operated by a computer integrated with the subsystem hardware. The subsystem computer performs specific functions directly related to the subsystem hardware. A workstation will be used to automate the operation of the tracking station and will provide a central facility to monitor the operation of the tracking station. The workstation and the subsystem controllers will be linked together through a Local Network Area (LAN). All of the software for these systems will be delivered as executable products. All of the systems will be installed and configured without software development, compilation and linking of code. The installation process will be automated to the greatest degree possible.

The operational scenario for this new station implements procedure control through the workstation. The workstation allocates the station resources required to support any given activity at the station. All high level control functions are initiated from the workstation. In turn, the workstation monitors the operation and performance of all the station subsystems and takes action to correct anomalies. Individual subsystems must initiate subordinate subsystem operations as required. And in turn, individual subsystems monitor the operation and performance of subordinate subsystems as necessary. In other words, all operation of the station is coordinated by the workstation, but individual subsystems will control and monitor other subsystems directly. Support files are managed by the workstation and transferred as required to the appropriate subsystem. The scenario outlined above encompasses the six basic functional requirements for monitor and control in the Deep Space Network tracking stations (see Table 1).

X-Windows

Several commercial companies are currently building tracking station components that provide an X-Window based Graphical User Interface (GUI) for operation of their equipment. Several NASA organizations have also provided an X-Windows based Graphical User Interface

(GUI) for operation of NASA developed equipment. Since X-Windows is a common solution to support remote operation of computers and in current use, we should examine its application as a standard for tracking station automation.

A tracking station built to be operated using the X-Windows protocol would require each subsystem to be designed as an x-client. In the example tracking station, each subsystem controller would come equipped with a GUI to support its operation. The Station Operations Workstation would be used as an x-server to operate each subsystem (see Figure 2). This approach permits the development of subsystems in isolation and safeguards the proprietary software of the commercial vendors. However, the X-Windows protocol was developed to support terminal operations on remote computers independent of the manufacturer. It was not designed to support automated operation of the remote computer. Consequently, an operator is required at the Station Operations Workstation to run the remote subsystems. In addition, X-Windows makes no provisions for the direct exchange of data between subsystems without operator involvement. The operational scenario requires subsystems to operate other subsystems and exchange data directly. Consequently, X-Windows alone will not fulfill the automation requirements.

Distributed Computing Environment

The emergence of the Open Software Foundation's (OSF) Distributed Computing Environment (DCE) has prompted speculation that DCE could be applied to the problems of tracking station automation. DCE was designed and developed to provide the services required by systems with multiple computers interconnected by a local area network (LAN) or a wide area network (WAN). As the name suggests, DCE services are designed to perform distributed computing. An underlying assumption for the development of DCE is that the work performed can be independent of location (that is, an application that requests a service is not concerned with where the service is performed). An overview of the DCE basic services with respect to the Open Systems Interconnect (OSI) Basic Reference Model is shown in Figure 3. There are five basic components to DCE:

1) The Distributed File Services (DFS) in DCE provide extensive tools to manage and manipulate files in a distributed computing system.

2) The DCE Time services provide for the synchronization of computer clocks in a distributed computing system.

3) The DCE Naming and Directory Services contains the names of users, machines and resources available in the distributed system

4) The DCE Management Services provide the tools to operate the distribute system.

5) The DCE Security Services control access to the distributed system.

All of these services use the DCE Remote Procedure Call (RPC) to access the network.

The application of DCE in a tracking station would likely rely heavily on the Remote Procedure Call (RPC) for most inter-processor communications. The DCE RPC provides an Interface Definition Language (IDL) which is used to create both client and server elements of an

RPC. The IDL also provides for the common representation of data in different computer systems. Once the IDL specification for an RPC is created, the IDL client and server elements are compiled and linked on the appropriate systems (see Figure 4). Applied to the example tracking station, each subsystem integrated into the station would include a set of client and server IDL definitions. The IDL definitions would be compiled and linked on the Station Operations Workstation. In addition, client and server IDL definitions would be compiled and linked on each subsystem that directly inter-operates with another subsystem. In the example tracking station, the application of the DCE RPC approach would produce the following scenario:

> A receiver subsystem is purchased and delivered along with a set of IDL specifications to support the operation of the receiver. The client IDL specifications are copied to the Station Operations Workstation, compiled and linked. Software is then developed to automate the receiver operation using the RPCs. In addition, the receiver operates as a client to access the antenna positions' values as a part of normal operations. The receiver also operates as a server to the antenna subsystem providing signal power measurements as required. The client and server IDL specifications for inter-operation of the receiver must be copied to the antenna subsystem, compiled and linked to support antenna-to-receiver communications. In turn, the antenna subsystem IDL specifications must be copied to the receiver subsystem, compiled and linked to support receiver-to-antenna communications.

A complex, highly automated tracking station would require hundreds (if not thousands) of RPCs to operate. Consequently, the management of RPCs will become a critical part of any DCE based tracking station. Though the DCE approach may offer a solution to the problems of inter-operability, compiling and linking RPCs from different vendors does not guarantee problem free integration. In addition, the DCE does not address the burden of software development for the Station Operations Workstation to automate the RPC functions.

The application of DCE Management Services (called Distributed Management Environment - DME) offers an alternative solution to compiling and linking IDL specifications into RPCs. The DME services provide high level data object management tools and are based on the Common Management Information Service Element (CMISE) standard. A DME based approach would be very similar to CMISE approach discussed in more detail in a later section.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) was developed in the Internet community to address the monitor and control of devices that support LANs and WANs. Network bridges and routers are typical devices where SNMP has be applied. To my knowledge, SNMP is not currently used or under consideration for use in tracking station operations. However, SNMP is similar to two protocols currently in use at tracking stations and is very similar to those protocols in its basic design. Therefore, a review of SNMP serves to identify common elements and functions in three similar protocols. In addition, deficiencies in the SNMP approach with respect to tracking station applications are identified.

SNMP provides a set of services designed to access the Management Information Base (MIB) established in a device. The MIB is a collection of objects that represent real resources in the device. For example, a network router used to bridge a local area network to an exterior communications line will have a network address and sub-network address. Each address can be an object in the network router MIB. The SNMP Get service provides for the retrieval of objects contained in a remote MIB. The SNMP Set service supports the modification of an object in a remote MIB. Also, SNMP has a Trap service that provides for a remote node to report a changed condition to a management node. In addition, software to access SNMP services through a GUI is available for workstations.

The application of SNMP in the example tracking station would find a Management Information Base installed on each subsystem (or device). The Station Operations Workstation would access each subsystem MIB using the SNMP Get and Set services (see Figure 5). The configuration and operation of subsystems would be accomplished using the Set service to change objects in the MIB. The status and performance of the subsystems would be determined by accessing objects in the MIB through the Get service. Anomalous conditions in the subsystems could be reported to the Station Operations Workstation using the Trap service. SNMP provides for the common representation of data through the Basic Encoding Rules (BER) to formulate messages in Abstract Syntax Notation 1 (ASN.1). SNMP services could also be used to support subsystem-to-subsystem communications. Using the antenna-to-receiver example discussed earlier, the receiver would use the Get service to access the antenna positions directly from the antenna subsystem. In turn, the antenna could use the Set service to initiate signal power measurements on the receiver and access the results using the Get service. Finally, commercial software to access SNMP services would be used to automate the Station Operations Workstation

There are however, a number of problems with the application of SNMP in a tracking station. First, SNMP Set and Get services are designed to operate on simple data types: scalars and two-dimensional arrays of scalars. Using SNMP Version 1, access to large sets of MIB data objects require multiple Sets or Gets. The SNMP GetNextRequest can simplify the process but this limitation still imposes performance constraints where large amounts of data must be accessed. SNMP Version 2 will expand the supported data types and add the GetBulkRequest service to address current limitations. Also, SNMP does not provide a service to access a directory to the contents of the MIB. The contents of the MIB can be determined through interrogation with a series of SNMP GetNextRequests, however: it is a time consuming process. A directory to the contents of the MIB is necessary to access specific data objects with Get and Set services. In addition, SNMP provides no mechanism to establish an alias data object. In the antenna-to-receiver example, the object names on both subsystems must match for the antenna or receiver to access each others MIB. For example:

> Company A builds the receiver with the name of the data object representing the operating radio frequency as “RF_Frequency”. Company B builds its telemetry processor with the same parameter represented with the name of “Operational_Frequency”. Under this condition, an SNMP Get made by the telemetry processor to access the receiver value of “RF_Frequency” would fail and generate an error.

A service to create an alias data object that could be associated with an existing data object would minimize the problems of inter-operation of subsystems. Finally, most implementations of SNMP operate over the User Datagram Protocol which is not a guaranteed delivery service. The successful operation of the tracking station will depend on the inter-subsystem communications. Consequently, a reliable protocol will be required to support the automation of the station.

The SNMP services were designed and developed to manage systems performing dedicated tasks in local and wide area networks. The functions performed by these systems are limited in scope and the services of SNMP reflect that limited scope. The subsystems in the tracking station also perform dedicated tasks; however, the scope of these tasks varies over a wide range of functions. The contents of each subsystem MIB will be completely different and a directory service would simplify the installation and management operations. This deficiency in SNMP could be addressed with implementation requirements imposed on the manufacturers. For example, a file with a directory to the MIB could be delivered with the product, copied to the Station Operations Workstation and made available to an application or user. Similarly, provisions could address the creation of alias named objects in remote MIBs. And, reliable transport services could be furnished by TCP. However, these implementation requirements amount to amendments to the SNMP specification which are unique requirements to the tracking station implementation.

Common Management Information Service Element

The successful implementation by the European Space Operations Center of a tracking station based on Common Management Information Service Element (CMISE) is a compelling rationale for further examination of this protocol. The Consultative Committee for International Telegraph and Telephone (CCITT) and the International Standards Organization (ISO) jointly developed CMISE as the management standard for equipment in the communications industry. The basic approach to the design of CMISE is similar to SNMP, however the eleven services provided by CMISE are more extensive and robust. Like SNMP, the services of CMISE are designed to manage data objects in a MIB. The CMISE Set and Get services are designed to operate on virtually any data type. Consequently, CMISE is not as limited as SNMP. In addition, the CMISE Event service is more robust and sophisticated than the SNMP Trap service. Like SNMP, CMISE provides for the common representation of data through the BER to formulate messages specified in ASN.1. And also like SNMP, CMISE provides no service to access a directory to the contents of the MIB. However, CMISE does provide Create and Delete services that could be used to establish alias data objects on remotes. For example:

> Company A builds the receiver with the name of the data object representing the operating radio frequency as “RF_Frequency”. Company B builds its telemetry processor with the same parameter represented with the name of “Operational_Frequency”. The telemetry processor would use the CMISE Create service to establish a data object called “Operational_Frequency” on the receiver and associated with the data object “RF_Frequency”. The receiver would then respond to a CMISE Get “Operational_Frequency”. The association of the two data objects would be part of the subsystem installation procedure. At the end of the activity, the telemetry processor would use the CMISE Delete service to remove “Operational_Frequency” from the MIB of the receiver.

The application of CMISE in the example tracking station, like SNMP, would find a Management Information Base installed on each subsystem (or device). The Station Operations Workstation would access each subsystem MIB using the CMISE Get and Set services (see Figure 5). The configuration and operation of subsystems would be accomplished using the Set service to change objects in the MIB. The status and performance of the subsystems would be determined by accessing objects in the MIB through the Get service. Anomalous conditions in the subsystems could be reported to the Station Operations Workstation using the CMISE Event service. CMISE services would also be used to support subsystem-to-subsystem communications. Finally, commercial software to access CMISE services would be used to automate the Station Operations Workstation. However, CMISE makes no provisions for file management. Consequently, an additional protocol will be required to move and manage the support files required to operate the subsystems and the station.

Manufacturing Message Specification

The process control protocol Manufacturing Message Specification (MMS) has also been successfully implemented in a tracking station. Originally sponsored by General Motors, MMS provides 86 services designed to support automation of factories. Like SNMP and CMISE, MMS is designed to manage the data objects in a MIB and provides for the common representation of data through the BER to formulate messages in ASN.1. And also like SNMP and CMISE, the systems managed through MMS perform dedicated tasks in the factory. However unlike SNMP or CMISE, MMS was designed to support systems that would span a wide range of manufacturing operations. Consequently, MMS provides 86 services to manage the resources in an automated facility.

The application of MMS in the example tracking station would find a Management Information Base installed on each subsystem (or device). The Station Operations Workstation would access each subsystem MIB using the MMS Read and Write services (see Figure 5). The configuration and operation of subsystems would be accomplished using the Write service to change objects in the MIB. The status and performance of the subsystems would be determined by accessing objects in the MIB through the Read service. Anomalous conditions in the subsystems could be reported to the Station Operations Workstation using the MMS Information Report service and the MMS Event Management services. MMS services would also be used to support subsystem-to-subsystem communications. Unlike CMISE, MMS provides an Identify service, GetCapabilityList service and a GetNamedVariableList service which describe the subsystem on request. The GetNamedVariableList service provides a directory to the contents of the MIB in the form of a list of the named objects contained in the MIB. The integration of different manufacturer's subsystems would be facilitated using the DefineNamedVariable and DeleteVariableAccess services to establish alias data objects on the station subsystems. Returning to the previous example:

> Company A builds the receiver with the name of the data object representing the operating radio frequency as "RF_Frequency". Company B builds its telemetry processor with the same parameter represented with the name of "Operational_Frequency". The telemetry processor would use the MMS DefineNamedVariable service to establish a data object called "Operational_Frequency" on the receiver and associated with the data object "RF_Frequency". The receiver would then respond to a MMS Read "Operational_Frequency". The association of the two data objects would be part of the subsystem installation procedure. At the end of the activity, the telemetry processor would use the DeleteVariableAccess service to remove "Operational_Frequency" from the MIB of the receiver.

Finally, the MMS file management services like FileOpen, FileRead, FileClose, FileDirectory, FileDelete and FileRename would be used to manage the support files required by the subsystems.

Beyond the basics, MMS provides services to support the kinds of subsystems commonly installed in tracking stations. The MMS Program Invocation Management services are designed to support subsystems with multi-tasking operating systems. Using MMS, a standard set of services can be used to start, stop, resume or kill programs running on remote subsystems without regard for the specifics of the target operating system. The Domain Management services support block memory transfers between subsystems. Using the MMS Domain services, subsystem configuration tables could be efficiently transferred between the Station Operations Workstation and the subsystems. The Journal Management services provide for the logging of activities and events in a process control environment. The Semaphore Management services provide support for systems with shared resources. In tracking stations with multiple antennas and limited equipment redundancy, contention for limited resources can be supported through MMS semaphore services.

An additional advantage to the employment of MMS, is the availability of “Application Enabler” products for use on the Station Operations Workstation to automate station operations. These products are commonly found in the manufacturing sector and often referred to as “Supervisory Control and Data Acquisition (SCADA)” packages. Used to automate factories, Application Enabler products are software packages that can be customized for a specific installation without software development. The companies that build Application Enablers provide communication drivers to access proprietary devices, like Programmable Logic Controllers (PLCs). Today, a number of these companies provide MMS communication drivers. Using these products in conjunction with MMS, the software for the Station Operations Workstation can be purchased and configured to operate the tracking station without software development.

Discussion

All five protocols surveyed could be used to build a spacecraft tracking station. However, each of these protocols were designed and developed for a specific environment. The question is “Which environment most closely matches to environment of a spacecraft tracking station?” A second question is “Which protocol will provide commercial vendors with the tools to develop and deliver products that can be installed and integrated without software development?”

Spacecraft tracking stations are composed of devices with dedicated resources performing dedicated operations. The antenna subsystem is dedicated to operating the antenna hardware while the receiver subsystem is dedicated to operating the receiver hardware. The operations performed by these subsystems vary significantly. X-Windows provides an environment for the remote operation of these devices but does not provide for automation. DCE provides an integration environment but does not relieve the burden of software development. The management of a device through its MIB with SNMP, CMISE or MMS can provide automation and relieve the burden of software development. However, the limited services of SNMP make it the least likely candidate for operation of a tracking station. Given the similarities between CMISE and MMS, what is the basis for a final selection? A detailed examination of these two protocols reveals some differences to direct a final selection.

At first glance, the CMISE Get and Set services appear nearly identical in function to the MMS Read and Write services. However, there are subtle differences between the two protocols that are derived from their intended applications. Consider the factory environment:

A factory is a confined environment where control must be decisive. Arbitrary control of a server might create catastrophic problems on the factory floor. Therefore, an MMS client must establish an association with a server before a dialog of MMS services can begin. If an association can not be established, control can not be initiated. Server systems are designed to fail in a safe mode, protecting the plant and personnel. When problems develop on the factory floor, MMS-based automation alerts operations personnel to investigate the problem and take corrective action. To provide decisive control, the exchange of MMS control messages employs confirmed services that require the client application receive an acknowledgment from the server application. The MMS Write service is a confirmed service that requires acknowledgment for completion.

Now consider a wide area communications network environment:

A wide area network is not a confined environment, frequently distributed over tens, or hundreds or thousands of miles. Communication device servers are also designed to fail in safe mode while redundancy provides for alternative means of communications. Rarely does a failure present a threat to life or property. Therefore, CMISE is designed to operate with or without an established association. The CMISE Set service can operate in both confirmed and unconfirmed modes.

The difference in these services in important for their respective applications. Corrective action in a factory frequently requires human intervention to safe guard life and property. Corrective action in a communications network can frequently be accomplished remotely. For example:

A recurring fault can cause a network router to fail. The router can be designed to reboot on failure to safe mode, reboot on failure to diagnostic mode or reboot on failure to operational mode. The recurring failure results in the router continuously rebooting. The time interval between faults is too short to support the normal establishment of an association and leading to a Set service to force the router into the diagnostic mode. The unconfirmed Set service provides a mechanism to reset the router to diagnostic mode before the fault occurs again.

Another subtle difference between CMISE and MMS can be seen in the Event services.

Both CMISE and MMS provide Event services. Though similar in principle, the services perform differently reflecting the environments for which they were designed. A detailed examination of the event data structures reveal that both CMISE and MMS provide an attribute for event-priority. However, only MMS provides an attribute for event-severity. From my experience, I believe this distinction is derived from the difference between the communications environment and the factory environment. Rarely do events in communications networks produce property or life threatening situations. However, events on the factory floor can produce these conditions. Therefore, the MMS Event service provides for severity of a failure.

Consequently, it is my opinion that MMS offers the best fit to the spacecraft tracking station environment. Based on experience, MMS provides the commercial vendors with a standard for automation. Using MMS, a commercial product can be installed and configured into an automate tracking station without site specific software development. And the availability of commercial products for factory automation based on MMS, supports this conclusion. In addition, MMS based Application Enabler products provide the tools to automate spacecraft tracking stations without traditional software development efforts.

References

Introduction to OSF DCE, Open Software Foundation, Prentice-Hall, Incorporated, 1992.

Schulz, Klaus-Jurgen; Service Management of a Spacecraft Ground Station Network in Support of Spacecraft Operations, Integrated Network Management, III (C-12), 1993 IFIP.

Stallings, William; SNMP, SNMPv2 and CMIP, The Practical Guide to Network-Management Standards, Addison-Wesley Publishing Company, 1993.

Tang, A. & Scoggins, S; Open Networking with OSI, Prentice-Hall, Incorporated, 1992.

International Organization for Standards, ISO 9506, Information Processing Systems - Open Systems Interconnection - Manufacturing Message Specification, 1989.

Heuser, Wm. Randy, Chen, Richard, and Stockett, Michael H.; Using Manufacturing Message Specification for Monitor and Control at Venus, Network Technology Conference, NASA Conference Publication 3241, 1993.

Table 1. This table provides a comparison of the functional requirements (down the left side) for monitor and control in Deep Space Network tracking stations and the protocols examined in this article (across the top).

Protocol Functional Requirements	**X-Windows**	**DCE**	**SNMP**	**CMISE**	**MMS**
Allocation of station resources	Yes	Yes	Yes	Yes	Yes
Configuration and Control of subsystems	Yes	Yes	Yes	Yes	Yes
Monitor status and performance	Yes	Yes	Yes	Yes	Yes
Inter-subsystem data exchange	No	Yes	Yes	Yes	Yes
Event and Alarm handling	No	No	Yes	Yes	Yes
Logging	No	No	No	Yes	Yes
File distribution and management	No	Yes	No	No	Yes
***No software development, compilation and linking**	Yes	No	Yes	Yes	Yes
***Data Object Alias**	No	No	No	Yes	Yes

* Derived requirement to support commercial products derived as executable products.

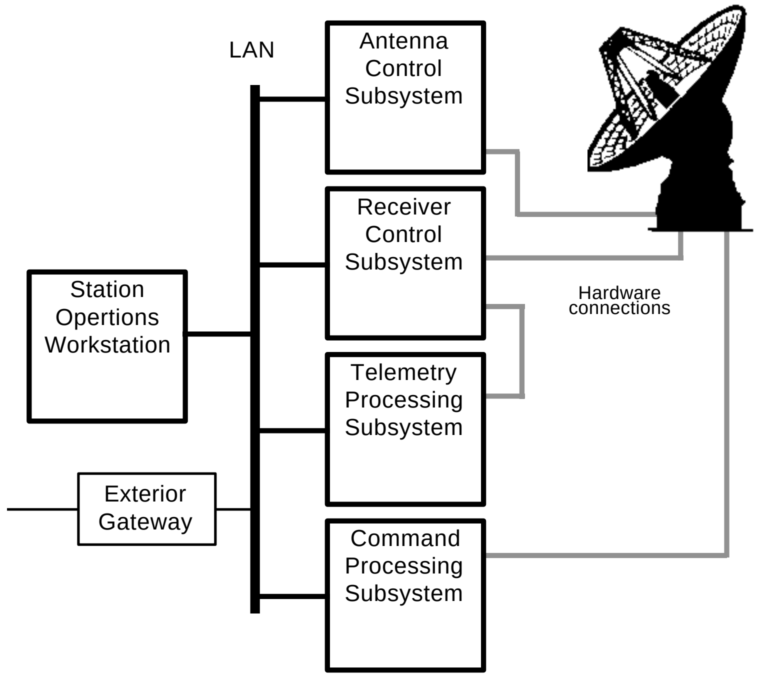

Figure 1. An example tracking station with four computer controlled subsystems inter-connected with a workstation through a Local Area Network.

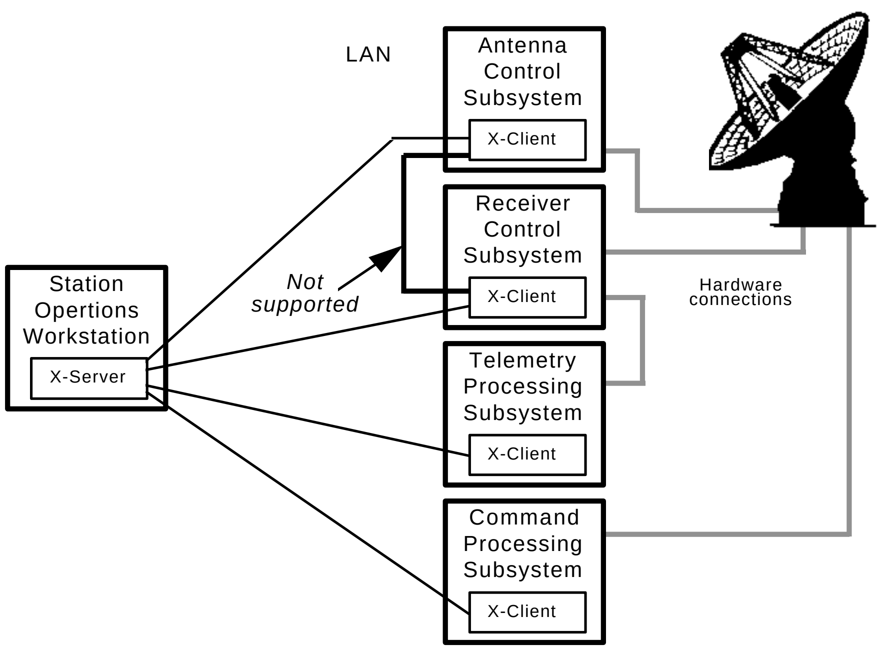

Figure 2. The application of X-Windows to support tracking station integration and automation would require each subsystem to operate as an x-client. The subsystems could be operated from the Station Operations Workstation operating as an x-client server. However, direct subsystem-to-subsystem data exchange is not supported by X-Windows.

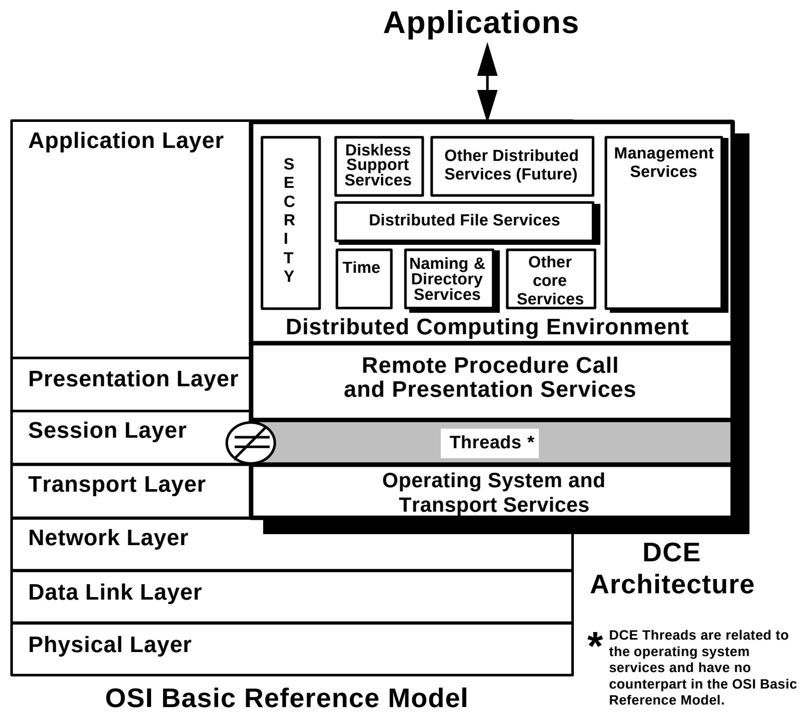

Figure 3. This figure shows the relationship between the DCE Architecture and the OSI Basic Reference Model.

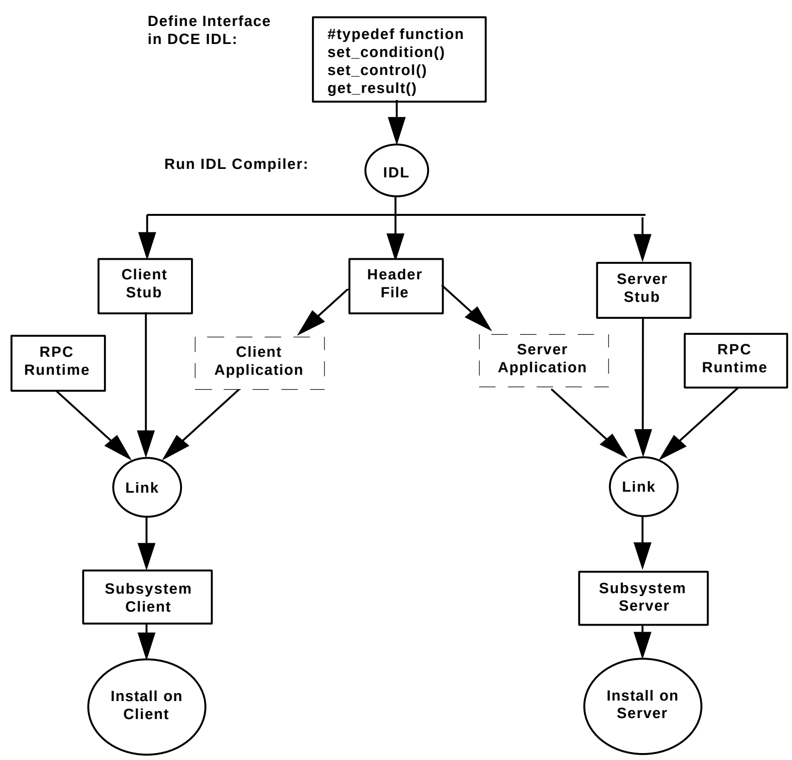

Figure 4. This figure shows the DCE process to create remote procedure calls from DCE IDL.

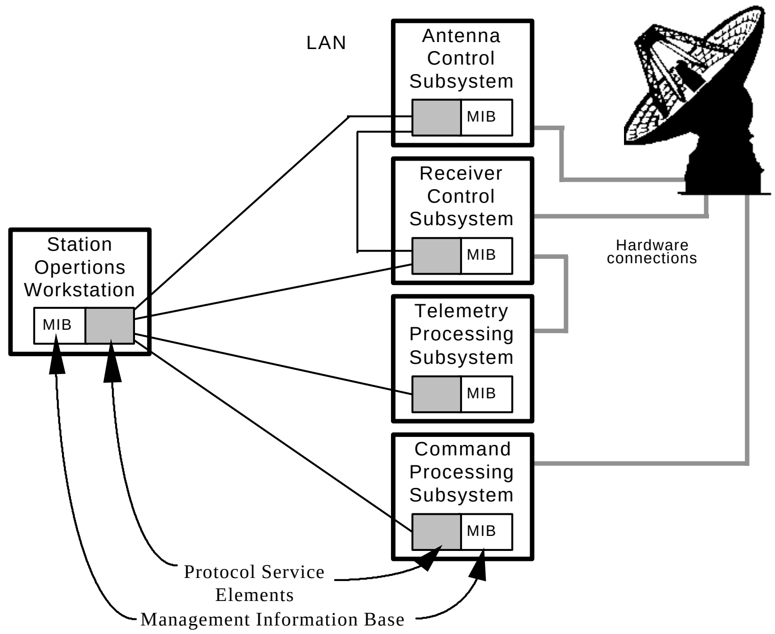

Figure 5. SNMP, CMIS/CMIP and MMS all provide services to access and manage a Management Information Base (MIB) on remote systems. In this example, the operator workstation provides monitor and control the of subsystems in a simple receiver only tracking station through services that access the MIB.